

## โจทย์ปัญหาเสริม

1. จงเขียนโปรแกรมเพื่อค้นหาชุดของเลขโดด 3 ตัวที่ไม่ซ้ำและเรียงลำดับจากน้อยไปมาก โดยมีเงื่อนไขว่า ผลรวมของเลขโดดทั้งสามนั้นมีค่ามากกว่า 10 โดยพิมพ์ผลลัพธ์ตามลำดับที่พบ หากพบชุดของตัวเลขที่ซ้ำก็ให้พิมพ์แค่ครั้งเดียว และหากไม่พบชุดของตัวเลขตามเงื่อนไขที่กำหนดให้พิมพ์ข้อความ NONE ดังตัวอย่าง

ข้อมูลนำเข้า	ผลลัพธ์
1246858594124665	246 468
124680585941666	246 468 058
235124	NONE

2. จงเขียนโปรแกรมเพื่อรับกลุ่มของตัวเลขเข้ามาเพื่อเปรียบเทียบหากกลุ่มตัวเลขโดดที่ติดกัน 3 ตัวและทั้งสามตัวนั้นต้องเรียงลำดับจากน้อยไปมากและมีค่ารวมกันได้มากกว่า 10 นำผลของกลุ่มตัวเลขโดดทั้งหมดที่หาได้มาแสดงเรียงลำดับจาก มากไปน้อย กรณีมีปรากฏซ้ำให้พิมพ์แสดงแค่ครั้งเดียว และในกรณีที่ไม่มีให้แสดงผลเป็น none ดังตัวอย่าง

ข้อมูลนำเข้าคือ:	ข้อมูลนำเข้าคือ:	ข้อมูลนำเข้าคือ:
52678589334467	933567946679	9332444
ผลลัพธ์คือ:	ผลลัพธ์คือ:	ผลลัพธ์คือ:
678	679	none
589	567	
467	356	
267		

3. จงเขียนโปรแกรมเพื่อนับจำนวนคำในประโยคที่กำหนดมาให้ (แต่ละประโยคจบด้วยจุด '.') โดยข้อมูล ที่อ่านเข้ามาในบรรทัดแรกคือคีย์เวิร์ดที่ใช้ในการค้นหา ตามด้วยประโยคที่ต้องการตรวจสอบ ดังตัวอย่าง

ตัวอย่างที่ 1 (กรณีมีค่าเท่ากับ to)

to  
Flu viruses which target man tend **to** attach **to** cells further up the airway, maximizing their chances of being passed on by coughing or sneezing.

ผลลัพธ์

2

ตัวอย่างที่ 2 (กรณีลงท้ายด้วย ing)

\*ing  
Flu viruses which target man tend to attach to cells further up the airway, **maximizing** their chances of **being** passed on by **coughing** or **sneezing**.

ผลลัพธ์

4

ตัวอย่างที่ 3 (กรณีขึ้นต้นด้วย Do โดยพิมพ์เล็กพิมพ์ใหญ่ให้ถือว่าเป็นตัวเดียวกัน)

Do\*  
**Don't** wait because you **don't** know how long it will take.

ผลลัพธ์

2

ตัวอย่างที่ 4 (กรณีขึ้นต้นด้วย th และลงท้ายด้วย e)

th\*e  
**The** return value of scanf() is **the** number of variables that were successfully assigned values, or EOF if **there** is an error.

ผลลัพธ์

3

หมายเหตุ สามารถมี \* ได้แค่ตำแหน่งเดียวในคีย์เวิร์ด

#### 4. Marking Expression

จงเขียนโปรแกรมเพื่อรับข้อความมาหนึ่งบรรทัด แล้วค้นหานิพจน์ทางคณิตศาสตร์ (arithmetic expressions) ทั้งหมดภายในข้อความที่รับมา เพื่อแทรกเครื่องหมาย  $\$...\$$  คร่อมนิพจน์เหล่านั้น เช่น

ข้อมูลนำเข้า:

Given  $\text{num1} + \text{num2} = -1$  and  $\text{num1} - \text{num2} = 5$ . What  $\text{num1}$  and  $\text{num2}$  are?

ข้อมูลส่งออก:

Given  $\$ \text{num1} + \text{num2} = -1 \$$  and  $\$ \text{num1} - \text{num2} = 5 \$$ . What  $\text{num1}$  and  $\text{num2}$  are?

ข้อมูลนำเข้า:

It can be concluded that if  $x > 0$ ,  $y > 0$  and  $z > 0$  then  $x + y + z > 0$ .

ข้อมูลส่งออก:

It can be concluded that if  $\$ x > 0 \$$ ,  $\$ y > 0 \$$  and  $\$ z > 0 \$$  then  $\$ x + y + z > 0 \$$ .

หมายเหตุ: นิพจน์ทางคณิตศาสตร์ของปัญหาข้อนี้จะเป็นกลุ่มของนิพจน์ที่มีเฉพาะเครื่องหมาย  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $=$ ,  $>=$ ,  $<=$ ,  $>$  และ  $<$  อยู่ภายใน เครื่องหมายอื่น ๆ นอกเหนือจากนี้จะไม่มีการ test cases

5. Shifting

จงเขียนโปรแกรมเพื่อดำเนินการขยับ (shift) ข้อมูลในอาร์เรย์ไปทางขวาเป็นจำนวน  $s$  ครั้ง โดยในอาร์เรย์มีจำนวนข้อมูลทั้งหมด  $n$  ค่า โดยแสดงผลลัพธ์สุดท้ายหลังการขยับ ดังตัวอย่าง

ลักษณะของข้อมูลนำเข้าบรรทัดแรก คือ จำนวนเต็ม  $n$  แทนจำนวนข้อมูลทั้งหมดในอาร์เรย์ บรรทัดที่สอง คือ รายการข้อมูลจำนวนเต็มทั้งหมด  $n$  ค่า บรรทัดที่สาม คือ ค่า  $s$  ซึ่งเป็นจำนวนเต็มที่มีค่ามากกว่าหรือเท่ากับ 0 ระบุจำนวนครั้งของการขยับข้อมูลในอาร์เรย์ไปทางขวา

ตัวอย่างที่ 1	ตัวอย่างที่ 2
ข้อมูลนำเข้า	ข้อมูลนำเข้า
5 //ค่า $n$	7
5 8 2 9 10 //ค่าข้อมูล $n$ ค่าที่ต้องวนรับ	5 3 8 2 9 11 4
2 //ค่า $s$ คือจำนวนการขยับ	3
ผลลัพธ์	ผลลัพธ์
9 10 5 8 2	9 11 4 5 3 8 2

## 6. Word Searching and Reversing

จงเขียนโปรแกรมเพื่อค้นหาคำ (key) และแทนที่คำที่พบด้วยการพิมพ์ย้อนกลับ จากรายการของคำทั้งหมด  $n$  คำ ( $0 < n < 1000$ ) ดังตัวอย่าง

Input:

person 9  
the person we love is the person we care

Output:

the nosrep we love is the nosrep we care 2

ข้อมูลนำเข้า บรรทัดแรก คือ คำที่ต้องการค้นหา (key) และจำนวนคำที่อยู่ในลิสต์ (n) บรรทัดถัดมา คือ รายการคำทั้งหมด  $n$  คำ

ผลลัพธ์ คือ ลิสต์ของคำทั้งหมดซึ่งรวมคำที่ถูกแทนที่เรียบร้อยแล้ว (ถ้ามี) ตามด้วยจำนวนคำที่พบ และถูกแทนที่ทั้งหมด (ตัวอย่างข้างบนนี้ คือ 2) แต่กรณีไม่เจอ พิมพ์ 0 ดังตัวอย่างข้างล่าง

Input:

Person 9  
the person we love is the person we care

Output:

the person we love is the person we care 0

\*หมายเหตุ พิมพ์ใหญ่พิมพ์เล็กให้ถือว่าต่างกัน

## 7. Evaluate Expression

จงเขียนโปรแกรมเพื่อคำนวณหาผลลัพธ์จากนิพจน์ทางคณิตศาสตร์ที่ประกอบด้วยตัวแปร  $x$  และ  $y$  และเครื่องหมายดำเนินการ  $+$ ,  $-$ ,  $*$  และ  $/$  โดยบรรทัดแรกรับนิพจน์การคำนวณ และบรรทัดที่สองรับค่าของตัวแปร  $x$  และ  $y$  หลังจากนั้นแสดงผลลัพธ์ที่ได้จากนิพจน์นั้น ดังตัวอย่าง

	ข้อมูลนำเข้า	ข้อมูลนำออก
ตัวอย่าง 1	$x - 2 + y$ 4 5	7.0
ตัวอย่าง 2	$x + 2 * y$ 4 -5	-6.0
ตัวอย่าง 3	$x + 2 * 2 - 6$ 4.5 0	2.5

### ข้อมูลนำเข้า

บรรทัดแรก คือ นิพจน์การคำนวณ ซึ่งมีเฉพาะตัวแปร  $x$ ,  $y$  (ตัวพิมพ์เล็ก) และค่าคงที่ภายในนิพจน์คำนวณสามารถมีตำแหน่งที่ว่าง (space) คั่นระหว่างตัวดำเนินการกับตัวดำเนินการหรือไม่ก็ได้ และสามารถมีตัวแปร  $x$  และ/หรือ  $y$  ได้หลายตำแหน่งหรือไม่เลยก็ได้ และนิพจน์ดังกล่าวไม่วงเล็บใด ๆ ภายใน

บรรทัดที่สอง คือ ค่าของตัวแปร  $x$  และ  $y$  ตามลำดับ เป็นจำนวนจริง

เครื่องหมายดำเนินการทางคณิตศาสตร์มีเฉพาะ  $+$ ,  $-$ ,  $*$  และ  $/$  ลำดับการคำนวณจะคำนวณตามลำดับความสำคัญของตัวดำเนินการ คือ ดำเนินการ  $*$  หรือ  $/$  ก่อน ดำเนินการ  $+$  หรือ  $-$  ที่หลัง และถ้าลำดับความสำคัญเท่ากันจะดำเนินการจากซ้ายไปขวา

ข้อมูลนำออก คือ ผลลัพธ์ของการดำเนินการ เป็นจำนวนจริงทศนิยม 1 ตำแหน่ง

หลักคิด: ใช้ *stack* ในการ *evaluate* หรือเก็บ *expression* ในลักษณะ *postfix* หรือ *prefix*

## 8. ลำดับตัวเลขที่หายไป (MissingNumInOrder)

กำหนดตัวเลขจำนวนเต็มสี่ตัวเมื่อเรียงลำดับ ผลต่างของตัวที่ติดกันจะมีค่าเท่ากัน เช่น

6 4 10 8 เมื่อเรียงลำดับจะได้ 4 6 8 10 ซึ่งผลต่างของตัวเลขที่อยู่ติดกันคือ 2 เท่ากันทั้งหมด

จากเงื่อนไขข้างต้น จงเขียนโปรแกรมหาตัวเลขตัวที่สี่ที่ขาดไป เมื่อมีข้อมูลนำเข้าเป็นตัวเลขจำนวนเต็ม 3 จำนวน โดยตัวเลขที่หายไปอาจอยู่ลำดับใดก็ได้ และตัวเลขที่หายไปต้องมีความมากกว่าตัวเลขที่มีค่าต่ำสุด

**ข้อมูลนำเข้า** คือ เป็นจำนวนเต็ม 3 จำนวน ที่มีอยู่ในช่วงตั้งแต่ -1000 ถึง 1000 โดยเขียนอยู่ในบรรทัดเดียวกัน คั่นด้วยหนึ่งช่องว่าง

**ข้อมูลส่งออก** คือ ตัวเลขที่หายไป ซึ่งมีความมากกว่าตัวเลขที่มีค่าต่ำสุด

ตัวอย่าง	ข้อมูลนำเข้า	ข้อมูลส่งออก
1	4 6 8	10
2	10 1 4	7
3	-4 4 8	0
4	0 0 0	0

1 4 1

## 9. Number Concatenation

ต่อชุดตัวเลข ข้อมูลนำเข้าบรรทัดแรกคือ จำนวนพจน์ของตัวเลข (n) สามหลักทั้งหมด และจำนวนพจน์ (m) ที่ต้องนำมาต่อกัน บรรทัดที่สองคือกลุ่มตัวเลขจำนวน n พจน์ ผลลัพธ์คือ ชุดของตัวเลขที่สามารถต่อกันได้เป็นวงจรเรียงจากน้อยไปมากซึ่งอาจมีได้มากกว่าหนึ่งชุด โดยวิธีการต่อ คือ ค้นหากลุ่มตัวเลขที่สามารถเรียงต่อกันได้เป็นวงจร เช่น 456 645 และ 554 สามารถต่อกันได้เพราะพจน์แรกลงท้ายด้วย 6 และพจน์สองขึ้นต้นด้วย 6 เช่นเดียวกับ 645 ต่อด้วย 554 ได้ โดยต้องต่อกันได้รวม m พจน์ และวงจรที่ได้จากการต่อขึ้นตำแหน่งแรกของพจน์แรกกับตำแหน่งสุดท้ายของพจน์สุดท้ายต้องเหมือนกัน และหากไม่สอดคล้องเงื่อนไขหรือไม่สามารถต่อได้ให้แสดงผลเป็น none ดังตัวอย่าง

ข้อมูลนำเข้า

5 3

345 456 223 554 645

ข้อมูลส่งออก

456 554 645

\*\*\* เนื่องจากพบวงจรในกลุ่มตัวเลขที่นำเข้ามา ดังนี้

456 645 554

554 456 645

645 554 456

\*\*\* 345 554 456 เคสนี้ต่อกันได้แต่ไม่ใช่วงจรขนาด 3

ข้อมูลนำเข้า

5 2

345 456 223 554 645

ข้อมูลส่งออก

none

ข้อมูลนำเข้า

6 2

345 456 923 644 242 232

ข้อมูลส่งออก

//ตัวอย่างนี้คำตอบมี 2 วงจร

232 242

456 644



## 10. Image Filtering

เขียนโปรแกรมเพื่อรับค่าขนาดภาพ จำนวนแถว  $\times$  จำนวนหลัก ตามด้วยข้อมูลภาพ แต่ละบรรทัด  
ซึ่งประกอบด้วยค่าข้อมูลแต่ละพิกเซลเป็น 0 หรือ 1 หลังจากนั้นให้ตรวจสอบข้อมูลภาพและ  
แสดงผลลัพธ์ออกมาเฉพาะตำแหน่งพิกเซลที่มีค่าเป็น 1 ดังตัวอย่าง

ข้อมูลนำเข้า:

```
5 9
000010000
000111000
011010110
000111000
000010000
```

ข้อมูลส่งออก:

```
1
111
11 1 11
111
1
```

## 11. Compression

ข้อมูลนำเข้า: AAABBBBDDDDDDDD

ข้อมูลส่งออก: 3A4B7D

ข้อมูลนำเข้า: ABBBBB

ข้อมูลส่งออก: A4BD

## 12. De-compression

ข้อมูลนำเข้า: 3A4B7D

ข้อมูลส่งออก: AAABBBBDDDDDDDD

ข้อมูลนำเข้า: 12A4BD

ข้อมูลส่งออก: AAAAAAAAAAABBBBD

### 13. The 3rd order

จงเขียนโปรแกรมเพื่อคำนวณหาข้อมูลลำดับที่ 3 จากผลการเรียงลำดับข้อมูลจากน้อยไปมาก ดังตัวอย่าง เมื่อ

ข้อมูลนำเข้า: บรรทัดแรก คือ จำนวนข้อมูล ( $n$ ) ซึ่งเป็นเลขจำนวนเต็มในช่วงตั้งแต่ 3 ถึง 100 และ  
บรรทัดถัดไป  $n$  บรรทัด คือ ข้อมูลที่เป็นจำนวนเต็มซึ่งอาจเรียงลำดับจากน้อยไปมากหรือไม่ก็ได้

ข้อมูลส่งออก: จำนวนเต็มลำดับที่ 3 ที่หายไป

ข้อมูลนำเข้า:

5  
12  
2  
13  
4  
9

ข้อมูลส่งออก:

9

#### 14. Safety Lock (V1)

เพื่อความปลอดภัยของกระเป๋าดำเนินทางจากการเปิดใช้โดยไม่ได้รับอนุญาต จึงพัฒนาระบบล็อกโดยใช้จำนวนเต็ม 3 หลักเป็นรหัสลับเพื่อการล็อกและเปิดกระเป๋าดำเนินทาง การเปิดจะใช้วิธีการหมุนเลขทั้งสามหลักให้ตรงกับรหัสล็อกที่ผู้ใช้ตั้งไว้เพื่อปลดล็อก

จึงเขียนโปรแกรมคำนวณและตรวจสอบการปลดล็อกกระเป๋าดำเนินทางเมื่อผู้ใช้หมุนตัวล็อกแต่ละหลัก ซึ่งอาจหมุนไปหน้า (forward) หรือย้อนกลับ (backward) ได้ตามจำนวนที่ต้องการ

##### ข้อมูลนำเข้า

- 0 3 7 คือ หมายเลขปัจจุบันของตัวล็อกกระเป๋าดำเนินทาง ซึ่งประกอบด้วย 3 ตำแหน่ง (หลัก) มีค่าในช่วง 0 ถึง 9
- 3 0 3 คือ รหัสปลดล็อกกระเป๋าดำเนินทาง แต่ละค่าเป็นเลขโดดในช่วง 0 ถึง 9
- B 21 คือ วิธีการหมุนหลักที่ 1 (B คือ หมุนย้อนกลับ) และจำนวนครั้งในการหมุน
- F 10 คือ วิธีการหมุนหลักที่ 2 (F คือ หมุนไปข้างหน้า) และจำนวนครั้งในการหมุน
- B 21 คือ วิธีการหมุนหลักที่ 3 และจำนวนครั้งในการหมุน

##### ข้อมูลนำออก

- 9 3 6 N คือ ผลลัพธ์ ประกอบด้วยตำแหน่งเลขปัจจุบันของตัวล็อกหลังการหมุนแต่ละหลัก และผลการปลดล็อก (O หรือ N)

#### 15. Safety Lock V2

##### ข้อมูลนำเข้า

- 0 3 7 คือ หมายเลขปัจจุบันของตัวล็อกกระเป๋าดำเนินทาง ซึ่งประกอบด้วย 3 ตำแหน่ง (หลัก) มีค่าในช่วง 0 ถึง 9
- 3 0 3 คือ รหัสปลดล็อกกระเป๋าดำเนินทาง แต่ละค่าเป็นเลขโดดในช่วง 0 ถึง 9
- 5 คือ จำนวนครั้งของการหมุน ( $0 \leq n \leq 100$ )
- 1 B 21 คือ หลักที่หมุน วิธีการหมุน (B คือ หมุนย้อนกลับ F คือ หมุนไปข้างหน้า) และจำนวนครั้งในการหมุน
- 2 F 10
- 3 B 21 รวม n บรรทัด
- 2 F 1
- 2 B 4

##### ข้อมูลนำออก

- 9 0 6 N คือ ผลลัพธ์ ประกอบด้วยตำแหน่งเลขปัจจุบันของตัวล็อกหลังการหมุนแต่ละหลัก และผลการปลดล็อก (O หรือ N)

## 16. Score Ranking

จึงเขียนโปรแกรมเพื่อประมวลผลการแข่งขันฟุตบอล โดยรับค่าข้อมูลการแข่งขัน คำนวณค่าสถิติ และ  
คะแนนสะสม เพื่อแสดงตารางสรุปผล ดังตัวอย่าง

ข้อมูลนำเข้า

3	คือ จำนวนทีม
A B C	คือ รายชื่อทีม เป็นอักษร A-Z
6	คือ จำนวนผลการแข่งขัน (ตามด้วยผลการแข่งขัน 6 รายการ)
A B 2 2	ผลรายการ 1 ประกอบด้วย ชื่อทีมเหย้า (A) ชื่อทีมเยือน (B) ประตูได้ (2) เสีย (2)
A C 3 1	
C B 0 1	
C A 1 0	
B C 3 1	
B A 0 0	

ข้อมูลส่งออก                      สรุปผลการแข่งขันเรียงลำดับตามคะแนน (Points) และจำนวนประตูได้เสีย (Gain)

B 2 2 0 3 8	ประกอบด้วย Team Win Draw Loss Gain Points
A 1 2 1 1 5	
C 1 0 3 -4 3	

(1) การคิดคำนวณแต้มสะสม จะคำนวณจากผลการแข่งขันแต่ละครั้งมารวมกัน โดยมีเงื่อนไข ดังนี้

- ถ้าการแข่งขันครั้งนั้นมีผลแพ้/ชนะ ทีมชนะจะได้ 3 แต้ม ในขณะที่ทีมแพ้ได้ 0 แต้ม
- ถ้าผลเป็นเสมอ จะได้ทีมละ 1 แต้ม

(2) การคำนวณจำนวนประตูได้เสีย จะคิดจากผลการแข่งแต่ละครั้ง เช่น ถ้าผลการแข่งขันครั้งหนึ่งเป็น

B C 3 1

จำนวนผลต่างประตูได้เสียของทีม B และ C ได้ในครั้งนี้ คือ 2 และ -2 ตามลำดับ (ทีมชนะได้บวกเท่ากับส่วน  
ต่างประตู ในขณะที่ทีมแพ้ลบเท่ากับส่วนต่างประตู)

ตัวอย่าง รายการสถิติการแข่งขัน (จาก output) เช่น B 2 2 0 3 8

หมายถึงทีม B ชนะมาแล้ว 2 ครั้ง เสมอ 2 ครั้ง แพ้ 0 ครั้ง จำนวนประตูได้เสียสะสม 3 ประตู และมี  
แต้มสะสม 8 แต้ม

## ตัวอย่าง 2

ข้อมูลนำเข้า:

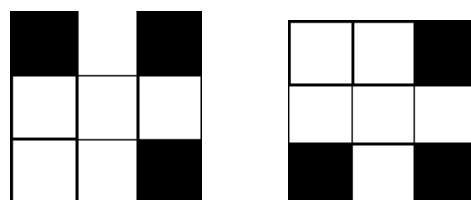
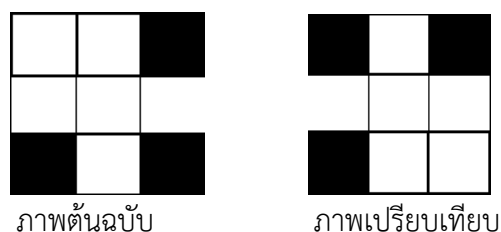
4  
A B C D  
2  
A B 2 2  
A C 0 1

ข้อมูลส่งออก:

C 1 0 0 1 3  
B 0 1 0 0 1  
A 0 1 1 -1 1  
D 0 0 0 0 0

## 17. Image Matching

ตรวจสอบความเหมือนกันของภาพขนาด  $n$  คูณ  $n$  ( $3 \leq n \leq 100$ ) สองภาพ ซึ่งหาผลการเปรียบเทียบครั้งแรกไม่เหมือนกัน สามารถหมุนภาพที่นำมาเปรียบเทียบตามเข็มนาฬิกา 90 องศา เพื่อเปรียบเทียบรอบถัด ๆ ไปได้ จนกว่าจะพบว่าเป็นภาพที่เหมือนกันหรือไม่ ดังตัวอย่าง



หมุนตามเข็มนาฬิกาที่ 1      หมุนตามเข็มนาฬิกาที่ 2 (ได้ผลเหมือนภาพต้นฉบับ)

ข้อมูลนำเข้า: บรรทัดแรก คือ ขนาดภาพ ( $n$  คูณ  $n$ ) ตามด้วยภาพต้นฉบับ  $n$  บรรทัด และ  $n$  บรรทัดถัดไป คือ ภาพที่ต้องการเปรียบเทียบ ระหว่างตัวเลขที่ว่างคั่น 1 space ขนาดของภาพสองภาพเท่ากันเสมอ

ตัวอย่าง 1 (ภาพข้างบน)	ตัวอย่าง 2	ตัวอย่าง 3	ตัวอย่าง 4
3 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 <div>             ข้อมูลภาพต้นฉบับ              ข้อมูลภาพเปรียบเทียบ           </div>	3 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 <div>             ข้อมูลภาพต้นฉบับ              ข้อมูลภาพเปรียบเทียบ           </div>	4 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 <div>             ข้อมูลภาพต้นฉบับ              ข้อมูลภาพเปรียบเทียบ           </div>	4 0 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 1 1 0 1 1 0 1 1 0 0 0 1 <div>             ข้อมูลภาพต้นฉบับ              ข้อมูลภาพเปรียบเทียบ           </div>
matched 2	not	matched 0	matched 1

ผลลัพธ์: หากสองภาพเหมือนกันให้แสดงข้อความว่า matched ตามด้วยจำนวนครั้งของการหมุนตามเข็มนาฬิกาจนกระทั่ง matched แต่หากไม่เหมือนกันให้แสดงข้อความ not

## 18. เกมสื่้อคำ

เกมสื่้อคำเกมหนึ่ง กำหนดให้ผู้เล่นเลือกคำภาษาอังกฤษแต่ละคำมาต่อกัน โดยตัวอักษรทุกตัวในแต่ละคำต้องเป็นสระในภาษาอังกฤษ A E I O หรือ U และคำที่เลือกได้ต้องนำมาจากพจนานุกรม ที่กำหนดให้เท่านั้น การเล่นเกมเริ่มโดยผู้เล่นเลือกคำจากพจนานุกรมมาหนึ่งคำ จากนั้นเลือกคำต่อไปโดยตัว อักษรตัวแรกของคำใหม่ต้องเป็นตัวเดียวกับตัวอักษรสุดท้ายของคำก่อนหน้า คำที่ถูกเลือกใช้แล้วไม่สามารถใช้ซ้ำได้อีก ทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งผู้เล่นไม่สามารถหาคำใด ๆ ในพจนานุกรมมาต่อได้อีก จงเขียนโปรแกรมหาจำนวนตัวสระที่เรียงต่อกันที่มากที่สุดที่เป็นไปได้ในการเล่นเกมหนึ่ง ๆ เมื่อกำหนดพจนานุกรมมาให้

### ข้อมูลนำเข้า (input)

บรรทัดแรก ประกอบด้วยจำนวนเต็ม  $N$  โดย  $N$  คือ จำนวนคำทั้งหมดในพจนานุกรม และ  $1 \leq N \leq 16$  บรรทัดถัดไป อีก  $N$  บรรทัด แต่ละบรรทัดแสดงแต่ละคำที่อยู่ในพจนานุกรมและมีความยาวไม่เกิน 100 ตัวอักษร คำทุกคำต้องสร้างจากตัวสระในภาษาอังกฤษตัวพิมพ์ใหญ่เท่านั้น (A, E, I, O, U) และคำทุกคำห้ามซ้ำกัน

### ข้อมูลส่งออก (output)

จำนวนตัวอักษรที่มากที่สุดที่เป็นไปได้ ในการใช้คำในพจนานุกรมที่ให้มาเรียงต่อกันตามเงื่อนไข

### ตัวอย่าง

input	input	input
3	4	5
AEIOU	AEEEE	IOO
UIU	OEOAEEE	IUUO
EO	AO	AI
	O	OIOOI
output	output	AOOI
8	13	output
		16

## 19. รถโดยสาร (Bus Time Table)

การเดินทางระหว่างเมือง A และเมือง B ต้องอาศัยรถโดยสารซึ่งเดินทางโดยไม่หยุดแวะระหว่างทาง เวลาที่น้อยที่สุดในการเดินทางระหว่างทั้งสองเมืองคือ  $J$  นาที และเวลาที่ผู้โดยสารขึ้นและลงรถรวมทั้งสองเมืองใช้เวลาอย่างน้อย  $K$  นาที ดังนั้นในการเดินทางครั้งหนึ่งต้องใช้เวลาอย่างน้อย  $J+K$  นาที หากรถโดยสารมาจากเมือง A ถึงเมือง B ก่อนหรือเท่ากับเวลาที่กำหนดจงเขียนโปรแกรมหาว่าต้องใช้รถโดยสารอย่างน้อยที่สุดกี่คันจึงเพียงพอต่อการให้บริการ เมื่อกำหนดเวลาออกจากแต่ละเมืองมาให้ โดยรถโดยสารทุกคันจะออกตรงเวลาเท่านั้น

**ข้อมูลนำเข้า (input):** บรรทัดแรก ประกอบด้วยตัวเลขจำนวนเต็ม  $J$  และ  $K$  ซึ่ง  $1 \leq J, K \leq 1000$  โดย  $J$  คือเวลาที่น้อยที่สุดที่ใช้ในการเดินทางระหว่างสองเมือง (หน่วยเป็นนาที) และ  $K$  คือเวลาที่น้อยที่สุดที่ผู้โดยสาร ขึ้นลงรถรวมกันทั้งสองเมือง (หน่วยเป็นนาที)

บรรทัดที่สอง แสดงถึงจำนวนเต็มบวก  $P$  ซึ่ง  $1 \leq P \leq 1440$  โดย  $P$  คือจำนวนตารางเวลาที่รถโดยสารออกจากเมือง A

บรรทัดถัดมาอีก  $P$  บรรทัด แต่ละบรรทัดแสดงเวลาที่รถโดยสารออกจากเมือง A โดยเรียงตามลำดับตามเวลา

บรรทัดถัดมา แสดงถึงจำนวนเต็มบวก  $Q$  ซึ่ง  $1 \leq Q \leq 1440$  โดย  $Q$  คือ จำนวนตารางเวลาที่รถโดยสารออกจากเมือง B

บรรทัดถัดมาอีก  $Q$  บรรทัด แต่ละบรรทัดแสดงเวลาที่รถโดยสารออกจากเมือง B โดยเรียงตามลำดับเวลา โดยเวลาอยู่ในรูปแบบของ HH:MM โดย HH เป็นชั่วโมง และ MM เป็นนาที หากเป็นหลักหน่วยต้อง ใส่ศูนย์นำหน้าเสมอ (เช่น 01:07) และเวลาต้องอยู่ระหว่าง 00:00 ถึง 23:59

**ข้อมูลส่งออก (output):** แสดงจำนวนรถโดยสารที่น้อยที่สุดที่ต้องใช้เพื่อให้การเดินทางระหว่างเมือง A และ B เป็นไปตาม ตารางเวลาที่ให้มา

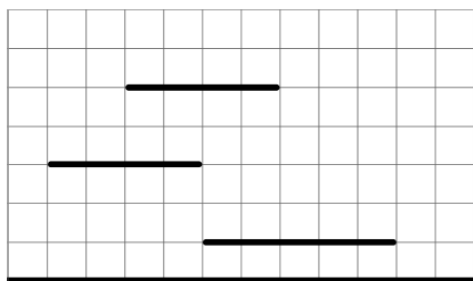
ตัวอย่าง

input	input	input
30 15	15 30	90 30
1	2	2
08:00	08:00	09:00
1	12:00	10:00
08:00	1	4
	08:45	08:00
output	output	11:00
2	1	14:00
		20:00
		output
		3



## 20. PLATFORM

พิจารณารูป A และ B



A



B

จากรูป A วาง ไม้กระดาน 3 แผ่น สูงจากพื้นห้อง 1, 3 และ 5 หน่วย รูป B วางเสารองรับไม้กระดานทั้ง 3 แผ่น โดยที่เสาอยู่ห่างจากปลายไม้กระดาน  $\frac{1}{2}$  หน่วยเสมอ ซึ่งจะต้องใช้เสาทั้งหมด 14 หน่วย  
จึงเขียนโปรแกรมหาความยาวที่น้อยที่สุดของเสาที่ใช้รองรับไม้กระดาน

**Input:** บรรทัดแรก คือ จำนวนไม้กระดาน  $N$  แผ่น ( $1 \leq N \leq 100$ )

$N$  บรรทัดต่อไป ประกอบด้วยจำนวนจริง 3 จำนวน คือ  $Y, X1, X2$  เมื่อ  $Y$  คือ ความสูงของไม้กระดานกับพื้นห้อง  $X1, X2$  คือ ตำแหน่งหัวและปลายไม้กระดาน โดยที่  $1 \leq Y, X1, X2 \leq 10000$  และ  $X2 > X1 + 1$  เสมอ (นั่นคือ ไม้กระดานจะยาวอย่างน้อย 2 หน่วย) และจะวางไม้กระดานสองแผ่นซ้อนกันโดยที่ไม่มีเสารองรับไม่ได้

**Output:** ความยาวที่น้อยที่สุดของเสาทั้งหมดที่ใช้รองรับไม้กระดานทุกแผ่น

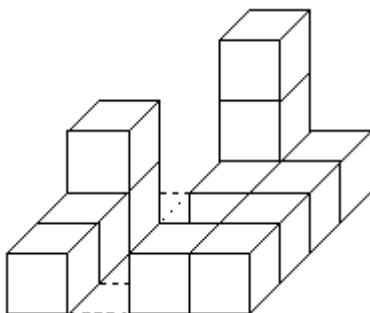
ตัวอย่าง

<b>Input</b> 3 1 5 10 3 1 5 5 3 7  <b>Output</b> 14	<b>Input</b> 5 50 50 90 40 40 80 30 30 70 20 20 60 10 10 50  <b>Output</b> 200
--	---

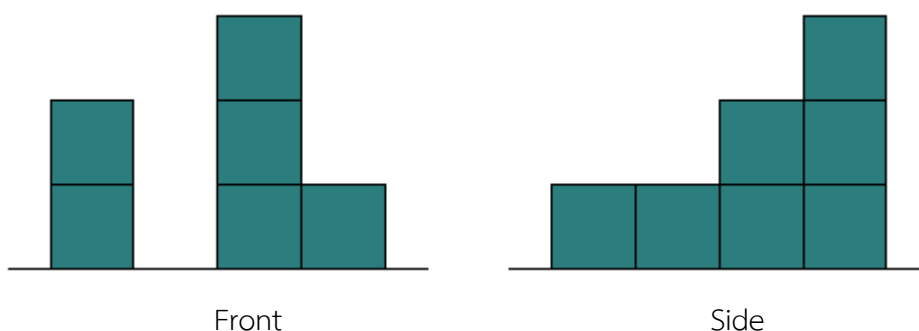
## 21. Cube Chaos

### Description

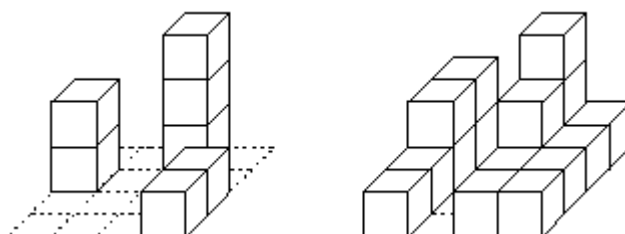
A construction is a number of cubes arranged on a square base as in the diagram below.



This is a four by four square supporting stacks of cubes. No stack will contain more than 8 cubes. The front and side projections of the construction are shown below.



Obviously there are a number of constructions that will satisfy both these projections. The constructions with the maximum and minimum number of blocks that will satisfy both projections are shown below.



Here the maximum number of blocks is 17, and the minimum is 7. Your job is to find the maximum and minimum number of blocks given the front and side projections.

## Input

The first line contains  $k$ , ( $1 \leq k \leq 8$ ), the length of the side of the square base. The next two lines contain descriptions of the construction (the front projection followed by the side projection). Each description consists of  $k$  integers. Each integer indicates the height of the corresponding projection of a stack of blocks.

## Output

The output will be one line containing the maximum number of blocks followed by a line containing the minimum number of blocks.

### Example 1

#### Input

```
4
2 0 3 1
1 1 2 3
```

#### Output

```
17
7
```

### Example 2

#### Input

```
1
1
1
```

#### Output

```
1
1
```

22. จงเขียนโปรแกรมเพื่อหาคำตอบของทุกความเป็นไปได้ของการจัดลำดับวงเล็บเปิด-ปิด จากวงเล็บเปิด-ปิดจำนวน  $n$  คู่ โดยลำดับการเปิด-ปิดเข้าหากันแต่ละคู่ต้องสมดุลหรือเป็นไปตามหลักของการวงเล็บเปิด-ปิดที่สามารถซ้อนกันเป็นชั้น ๆ ได้ ดังตัวอย่าง

**ข้อมูลนำเข้า** คือ จำนวนคู่ของวงเล็บ ( $n$ ) ซึ่งเป็นจำนวนเต็มบวกที่มากกว่าหรือเท่ากับ 1

**ข้อมูลนำออก** คือ แบบของการจัดวงเล็บเปิด-ปิด ที่เป็นไปได้ แยกแสดงเป็นบรรทัด ๆ ตามจำนวนแบบที่เป็นได้ทั้งหมด

**ตัวอย่างที่ 1**

```
1 //input กรณี n=1
() //output แบบวงเล็บที่เป็นไปได้
```

**ตัวอย่างที่ 2**

```
2 //input กรณี n=2
() () //output แบบวงเล็บที่เป็นไปได้ 1
(()) //output แบบวงเล็บที่เป็นไปได้ 2
```

**ตัวอย่างที่ 3**

```
3 //input กรณี n=3
() () () //output คือ แบบวงเล็บที่เป็นไปได้ทั้งหมด
() (())
(()) ()
((()))
```

เขียนแบบ recursive อาจช่วยให้เขียนง่ายขึ้น

## 23. Shifting Screen

จงเขียนโปรแกรมเพื่อขยับ (shift) ภาพต้นฉบับไปทางซ้ายหรือขวาของเฟรมแสดงผลตามที่กำหนด การ  
ขยับซ้ายหรือขวานั้น หากภาพสุดขอบเฟรมให้วนกลับมาแสดงต่อยังขอบอีกด้าน ดังตัวอย่าง

Example 1:

Input: 5 7	line 1 size of data
0 0 0 0 0 0 0	data of row 1
0 1 0 0 0 0 0	data of row 2
0 1 1 0 0 0 0	data of row 3
0 1 1 1 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5
L 2	shift left 2 steps
Output:	
0 0 0 0 0 0 0	data of row 1
0 0 0 0 0 0 1	data of row 2
1 0 0 0 0 0 1	data of row 3
1 1 0 0 0 0 1	data of row 4
0 0 0 0 0 0 0	data of row 5

Example 2:

Input: 5 7	line 1 size of data
0 0 0 0 0 0 0	data of row 1
0 1 0 0 0 0 0	data of row 2
0 1 1 0 0 0 0	data of row 3
0 1 1 1 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5
R 1	shift right 1 step
Output:	
0 0 0 0 0 0 0	data of row 1
0 0 1 0 0 0 0	data of row 2
0 0 1 1 0 0 0	data of row 3
0 0 1 1 1 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5

### ข้อมูลนำเข้า

บรรทัดที่ 1 ขนาดของข้อมูลนำเข้า แถว (m) x คอลัมน์ (n)

m แถวถัดมา คือ ค่าข้อมูลในแต่ละคอลัมน์ ซึ่งประกอบด้วย 0 หรือ 1 ซึ่งคั่นด้วยที่ว่าง 1 ตำแหน่ง

บรรทัดสุดท้าย คือ รูปแบบการขยับภาพ (L/R) และจำนวนครั้งของการขยับ (เป็นจำนวนเต็มบวก <= 1000) เช่น L 2 คือ ขยับไปทางซ้าย 2 ตำแหน่ง, R 1 คือ ขยับไปทางขวา 1 ตำแหน่ง

### ข้อมูลนำออก

ผลลัพธ์หน้าจอแสดงผลหลังดำเนินการขยับภาพตามที่กำหนด ซึ่งมีขนาดเท่ากับ m x n เช่นเดิม

หมายเหตุ การใช้เศษจากการหาร (modulo) จะช่วยให้เขียนโปรแกรมง่ายขึ้น

ภาควิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
มหาวิทยาลัยนเรศวร



วันที่ 4 พฤษภาคม 2554

### ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 9 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

#### อาหารโอชา (Food)

ในพระราชวังแห่งหนึ่ง พ่อครัวสามารถทำอาหารได้  $n$  ชนิดที่แตกต่างกัน อาหารเหล่านี้อยู่ในเซต  $F = \{f_1, f_2, f_3, \dots, f_n\}$  ในการถวายอาหารแก่เจ้าชายซึ่งกินจุมาก พ่อครัวจะถวายอาหารหนึ่งชนิดต่อหนึ่งชั่วโมง และจะถวายจนครบ  $n$  ชนิด ในการถวายอาหารนี้ พ่อครัวมีกลุ่มของอาหารที่ต้องห้ามไม่ให้ถวายเป็นลำดับแรก  $m$  ชนิด กำหนดอยู่ในเซต  $P \subset F$  ให้เขียนโปรแกรมแสดงลำดับทั้งหมดของการถวายอาหาร  $n$  ชนิด

#### ข้อมูลนำเข้า

- บรรทัดแรกเป็นจำนวนชนิดอาหาร  $n$  โดยที่  $2 \leq n \leq 8$
- บรรทัดที่สองเป็นจำนวนชนิดอาหารต้องห้าม  $m$  โดยที่  $2 \leq m < n$
- บรรทัดที่สามแสดงชนิดของอาหารต้องห้ามที่ไม่ให้ถวายเป็นลำดับแรก โดยแสดงเป็นตัวเลขจำนวนเต็มบวก  $m$  ตัว โดยมีช่องว่างคั่นอยู่ระหว่างตัวเลข

#### ข้อมูลส่งออก

แสดงลำดับที่เป็นไปได้ทั้งหมด โดยใช้หนึ่งบรรทัดในการแสดงลำดับของอาหารหนึ่งลำดับ ลำดับของอาหารจะเป็นตัวเลขจำนวนเต็มบวกระหว่าง 1 ถึง  $n$  ที่มีช่องว่างคั่นอยู่ระหว่างตัวเลข สำหรับการเรียงก่อนหลังของลำดับของอาหาร ไม่มีผลต่อความถูกต้องในการตรวจสอบผลลัพธ์

#### หมายเหตุ

- แนะนำให้ใช้ printf ในการแสดงผล
- ข้อมูลส่งออกที่ได้จากชุดทดสอบมีขนาดไม่เกิน 1 MB

ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
4	4 1 2 3
3	4 1 3 2
1 2 3	4 2 1 3
	4 2 3 1
	4 3 1 2
	4 3 2 1

ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
4	1 2 3 4
2	1 3 2 4
3 2	1 4 2 3
	1 2 4 3
	1 3 4 2
	1 4 3 2
	4 2 3 1
	4 3 2 1
	4 1 2 3
	4 2 1 3
	4 3 1 2
	4 1 3 2

### ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	16 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

### ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล food.c และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: food
```

```
LANG: C
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```

---

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล food.cpp และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: food
```

```
LANG: C++
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```



ภาควิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
มหาวิทยาลัยนเรศวร



วันที่ 5 พฤษภาคม 2554

### ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 8 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

#### คู่ตัวเลขเด่น (Pair)

ให้ชุดของคู่อันดับจำนวนเต็มบวกมา  $n$  ชุด คือ  $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)$  โดยที่  $a_i \neq a_j$  ถ้า  $i \neq j$  และ  $b_k \neq b_l$  ถ้า  $k \neq l$  โดยกำหนดว่า  $1 \leq a_i \leq 100,000$  และ  $1 \leq b_j \leq n$  เราเรียกคู่อันดับ 2 คู่  $(a_i, b_i)$  และ  $(a_j, b_j)$  ว่าคู่ตัวเลขเด่นก็ต่อเมื่อ  $a_i > a_j$  และ  $b_i < b_j$

จงเขียนโปรแกรมที่มีประสิทธิภาพในการหาค่าผลรวมของ  $a_i + a_j$  ทั้งหมด เมื่อ คู่  $(a_i, b_i)$  และ  $(a_j, b_j)$  เป็นคู่ตัวเลขเด่น

#### ข้อมูลนำเข้า

ข้อมูลนำเข้ามีลักษณะ ดังนี้

1. บรรทัดที่หนึ่งเป็นค่าของ  $n$  โดยที่  $2 \leq n \leq 100,000$

2. บรรทัดที่สองเป็นค่าของคู่ตัวเลข  $a_i$  และ  $b_i$  จำนวน  $n$  คู่ โดยจะเรียงจากคู่ที่หนึ่งไปจนกระทั่งถึงคู่ที่  $n$  โดยมีตัวเลขทั้งหมด  $2n$  ตัว และมีช่องว่างคั่นอยู่ระหว่างตัวเลขเหล่านี้

#### ข้อมูลส่งออก

เป็นตัวเลขจำนวนเต็มบวกหนึ่งค่า ซึ่งแสดงถึงผลรวมของ  $a_i + a_j$  ทั้งหมด เมื่อ คู่  $(a_i, b_i)$  และ  $(a_j, b_j)$  เป็นคู่ตัวเลขเด่น

#### หมายเหตุ

- แนะนำให้ใช้ scanf ในการรับค่าและ printf ในการแสดงผล
- แนะนำให้ใช้ตัวแปรชนิด double ในการเก็บค่าผลรวม และแสดงผลโดยใช้รูปแบบ "%.01f"

#### ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
6	78
2 1 7 6 9 3 18 4 3 2 5 5	

## ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
4	39
1 4 3 2 2 3 7 1	

## ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	2 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

## ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อเพิ่มข้อมูล pair.c และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
TASK: pair
LANG: C
AUTHOR: YourName YourLastName
CENTER: YourCenter
*/
```

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อเพิ่มข้อมูล pair.cpp และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
TASK: pair
LANG: C++
AUTHOR: YourName YourLastName
CENTER: YourCenter
*/
```



### ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 8 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

#### ระเบิดมหาประลัย (Bomb)

ทหารนาวิกโยธินกำลังต้องการที่จะบุกเข้าไปชิงตัวประกันออกมาจากสถานที่ลับแห่งหนึ่ง ในการที่จะบุกเข้าไปในที่แห่งนี้ ทหารนาวิกโยธินจะต้องผ่านเหมืองระเบิด โดยในเหมืองระเบิดนี้จะมีทั้งระเบิดจริงและระเบิดปลอมอยู่ทั้งหมดจำนวน  $n$  ตำแหน่งที่ไม่ซ้ำกัน คือ  $\{p_1, p_2, p_3, \dots, p_n\}$  โดยที่  $p_i = (x_i, y_i)$  เป็นพิกัดของระเบิด หน่วยข่าวกรองของทหารทราบมาว่า ระเบิดจริงจะอยู่ในตำแหน่งที่มีลักษณะพิเศษที่เรียกว่าตำแหน่งมหันตภัย ซึ่งลักษณะพิเศษดังกล่าวถูกระบุตามเงื่อนไขดังต่อไปนี้

1. ศัพท์ทางการทหารกล่าวว่าตำแหน่ง  $p_1$  บดบังตำแหน่ง  $p_2$  ก็ต่อเมื่อ  $x_1 > x_2$  และ  $y_1 > y_2$
2. ตำแหน่งมหันตภัยคือ ตำแหน่งที่ไม่มีตำแหน่งอื่น ๆ บดบัง

จงเขียนโปรแกรมที่มีประสิทธิภาพในการระบุตำแหน่งมหันตภัยที่มีระเบิดจริงทั้งหมด

#### ข้อมูลนำเข้า

1. บรรทัดแรกเป็นค่าของตัวแปร  $n$  โดยที่  $1 \leq n \leq 1,000,000$
2. บรรทัดที่สองถึง  $n+1$  ระบุตำแหน่งของระเบิดทั้งหมด แต่ละบรรทัดระบุค่าของตำแหน่งเป็นจำนวนเต็มบวกสองตัว  $x$  และ  $y$  โดยมีช่องว่างคั่นอยู่ระหว่างตัวเลขทั้งสอง โดยที่  $1 \leq x, y \leq 10,000,000$

#### ข้อมูลส่งออก

ให้ระบุตำแหน่งมหันตภัยทั้งหมด โดยให้แต่ละบรรทัดระบุค่าของตำแหน่งเป็นจำนวนเต็มบวกสองตัว  $x$  และ  $y$  โดยมีช่องว่างคั่นอยู่ การเรียงก่อนหลังของตำแหน่ง ไม่มีผลต่อความถูกต้องในการตรวจสอบผลลัพธ์

หมายเหตุ แนะนำให้ใช้ scanf ในการรับค่าและ printf ในการแสดงผล

ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
5	9 1
9 1	8 2
8 2	7 3
7 3	6 4
6 4	5 5
5 5	

ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
7	3 7
1 2	6 6
2 4	7 3
4 1	
7 3	
5 5	
6 6	
3 7	

ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	2 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

### ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล bomb.c และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: bomb
```

```
LANG: C
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```

---

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล bomb.cpp และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: bomb
```

```
LANG: C++
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```



ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 9 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

ขับรถหลบสิ่งกีดขวาง (Car)

ในการแสดงขับรถผาดโผนบนถนนที่มีเลนทั้งหมด  $m$  เลน โดยให้หมายเลขประจำเลนจากซ้ายไปขวามีค่าตั้งแต่ 1 จนถึง  $m$  ตามลำดับ นักแสดงขับรถผาดโผนต้องบังคับรถให้แล่นไปบนถนนดังกล่าวให้ปลอดภัยตลอดระยะเวลา  $t$  หน่วย การแสดงเริ่มต้น ณ เวลา  $t = 0$  นักแสดงขับรถผาดโผนอยู่ในเลนที่  $n$

ในแต่ละ 1 หน่วยเวลา อาจมีสิ่งกีดขวางตกลงมายังถนนบางเลน ทำให้เขาต้องบังคับรถเพื่อหลีกเลี่ยงสิ่งกีดขวาง ซึ่งมีทางเลือกในการบังคับรถอยู่ 3 แบบ ได้แก่ 1 หมายถึง การเปลี่ยนเลนไปทางซ้าย 1 เลนในเวลาถัดไป (ไปยังเลนที่มีหมายเลขประจำเลนน้อยกว่า), 2 หมายถึงการเปลี่ยนเลนไปทางขวา 1 เลนในเวลาถัดไป (ไปยังเลนที่มีหมายเลขประจำเลนมากกว่า) และ 3 หมายถึง การขับอยู่ในเลนเดิม กำหนดให้ถนนเป็นเส้นตรงตลอดทาง

จงเขียนโปรแกรมเพื่อบังคับให้รถแล่นไปตามเส้นทางนี้โดยปลอดภัย โดยชุดข้อมูลทดสอบจะมีคำตอบที่ถูกต้องเพียง 1 คำตอบเสมอ

ข้อมูลนำเข้า

- บรรทัดแรกระบุจำนวนเลน  $m$  โดยที่  $2 \leq m \leq 40$
- บรรทัดที่สองระบุหมายเลขเลนเริ่มต้น  $n$  โดยที่  $1 \leq n \leq m$
- บรรทัดที่สามระบุระยะเวลา  $t$  โดยที่  $1 \leq t \leq 100$
- บรรทัดที่สี่ถึงบรรทัดที่  $t + 3$  แสดงสถานะของถนน ณ เวลา  $1, 2, \dots, t$  ตามลำดับ แต่ละบรรทัดระบุตัวเลข  $m$  ตัว เลขแต่ละตัวแสดงสถานะของถนน ตั้งแต่เลนที่ 1 ถึงเลนที่  $m$  โดยเลข 0 หมายถึงเลนนั้นไม่มีสิ่งกีดขวาง และเลข 1 หมายถึงมีสิ่งกีดขวางอยู่

ข้อมูลส่งออก

มีอยู่  $t$  บรรทัด แต่ละบรรทัดมีตัวเลข 1 ตัวเพื่อแสดงถึงทางเลือกในการบังคับรถของนักแสดงขับรถผาดโผน ในแต่ละช่วงเวลา บรรทัดที่  $i$  หมายถึงการเปลี่ยนเลนจากเวลาที่  $i - 1$  ไปยังเวลาที่  $i$  เมื่อ  $i = 1, 2, \dots, t$  โดยที่เลข 1 จะหมายถึงขับไปทางซ้าย 1 เลน, เลข 2 หมายถึงขับไปทางขวา 1 เลน, และเลข 3 หมายถึงขับอยู่ในเลนเดิม

ตัวอย่างที่ 1

<u>ข้อมูลนำเข้า</u>	<u>ข้อมูลส่งออก</u>
7	1
5	1
5	1
0 0 0 0 0 0	1
0 0 0 0 0 0	2
0 0 0 0 0 0	
0 1 1 0 0 0	
1 0 1 1 1 1	

ตัวอย่างที่ 2

<u>ข้อมูลนำเข้า</u>	<u>ข้อมูลส่งออก</u>
5	2
2	2
3	3
0 0 0 1 0	
0 1 1 0 0	
1 1 1 0 1	

ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	2 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	16 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

### ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล car.c และระบุส่วนหัวของโปรแกรดังนี้

```
/*
```

```
TASK: car
```

```
LANG: C
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```

---

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล car.cpp และระบุส่วนหัวของโปรแกรดังนี้

```
/*
```

```
TASK: car
```

```
LANG: C++
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```



ภาควิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
มหาวิทยาลัยนเรศวร



วันที่ 5 พฤษภาคม 2554

### ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 8 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

#### สถานีอวกาศ (Space)

วิศวกรต้องการสร้างอาณานิคมแห่งหนึ่งในจักรวาล อาณานิคมนี้ประกอบด้วยสถานีอวกาศ (Space Station) ทั้งหมด  $2^d$  สถานี โดยมีเลขประจำสถานี  $d$  หลัก ที่ประกอบด้วยเลขศูนย์และเลขหนึ่งเท่านั้น เช่น ในกรณีที่  $d = 2$  จำนวนของสถานีทั้งหมดจะเท่ากับ 4 และมีเลขประจำสถานีคือ 00, 01, 10 และ 11 ในการสร้างอาณานิคมแห่งนี้ วิศวกรจะต้องสร้างเส้นทางเชื่อมระหว่างสถานี โดยมีกฎในการสร้างอยู่ว่าสถานีสองแห่งใด ๆ จะมีเส้นทางเชื่อมต่อกัน ก็ต่อเมื่อ เลขประจำสถานีของทั้งสองสถานีแตกต่างกันอยู่หนึ่งหลักพอดี

จงเขียนโปรแกรมในการสร้างเส้นทางเชื่อมเหล่านี้ โดยพิมพ์เส้นทางเชื่อมแต่ละเส้นทางเพียงครั้งเดียว สำหรับ ลำดับของสถานีในการแสดงเส้นทางเชื่อม และการเรียงก่อนหลังของการแสดงเส้นทางเชื่อม ไม่มีผลต่อความถูกต้อง ในการตรวจสอบผลลัพธ์

#### ข้อมูลนำเข้า

มีอยู่หนึ่งบรรทัดเป็นตัวเลขจำนวนเต็มบวกที่แสดงค่าของ  $d$  โดยที่  $2 \leq d \leq 14$

#### ข้อมูลส่งออก

มีอยู่  $d \times 2^{d-1}$  บรรทัด ที่แสดงเส้นทางเชื่อมทั้งหมด โดยให้แต่ละบรรทัดแสดงเส้นทางเชื่อมหนึ่งเส้น โดยแสดงเป็น เลขประจำสถานี  $d$  หลักสองสถานีที่มีเส้นทางเชื่อมกันอยู่ โดยมีช่องว่างคั่นระหว่างตัวเลขสองตัวนั้น

หมายเหตุ แนะนำให้ใช้ printf ในการแสดงผล

ตัวอย่างที่ 1

<u>ข้อมูลนำเข้า</u>	<u>ข้อมูลส่งออก</u>
2	00 01 00 10 01 11 10 11

ตัวอย่างที่ 2

<u>ข้อมูลนำเข้า</u>	<u>ข้อมูลส่งออก</u>
3	000 001 010 011 000 010 001 011 100 101 110 111 100 110 101 111 000 100 001 101 010 110 011 111

ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

### ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล space.c และระบุส่วนหัวของโปรแกรมดังนี้

```
/*  
TASK: space
```

```
LANG: C
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```

---

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล space.cpp และระบุส่วนหัวของโปรแกรมดังนี้

```
/*
```

```
TASK: space
```

```
LANG: C++
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: YourCenter
```

```
*/
```



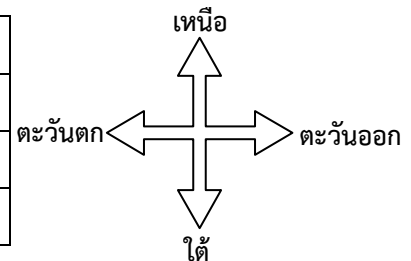
ข้อสอบการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 7

ข้อสอบมี 3 ข้อ 9 หน้า ให้ทำทุกข้อ เวลา 09:00 – 12:00 น.

**ท่ระบายน้ำ (Sewer)**

เมืองแห่งหนึ่งมีพื้นที่เป็นรูปสี่เหลี่ยมขนาด  $a$  แถวคูณ  $b$  คอลัมน์และแบ่งเขตเป็นจำนวนเท่ากับ  $a \times b$  เขต แต่ละเขตจะมีพิกัด  $(i, j)$  โดยเขตที่พิกัด  $(1, 1)$  จะอยู่ที่มุมซ้ายบนของพื้นที่สี่เหลี่ยม และแต่ละเขตจะมีท่ระบายน้ำเชื่อมต่อกับเขตเพื่อนบ้านหรือไม่ก็ได้ ดังแสดงในรูป (ให้เครื่องหมาย  $\updownarrow$  และ  $\leftrightarrow$  แสดงถึงท่ระบายน้ำที่เชื่อมระหว่างเขต)

(1, 1)	$\leftrightarrow$	(1, 2)	$\updownarrow$	(1, 3)		(1, 4)
(2, 1)		(2, 2)	$\leftrightarrow$	(2, 3)	$\leftrightarrow$	(2, 4)
(3, 1)	$\leftrightarrow$	(3, 2)	$\updownarrow$	(3, 3)	$\leftrightarrow$	(3, 4)
(4, 1)		(4, 2)		(4, 3)		(4, 4)



กำหนดให้เขตที่พิกัด  $(1, 1)$  เป็นจุดเริ่มปล่อยน้ำทิ้ง โดยจะสามารถระบายน้ำทิ้งไปยังท่ระบายน้ำที่เชื่อมอยู่กับเขตนั้นๆ และแต่ละท่ใช้เวลาระบายน้ำทิ้งจากเขตหนึ่งไปยังเขตหนึ่งด้วยเวลาหนึ่งหน่วย น้ำสามารถไหลได้ 4 ทิศทาง คือ ไหลไปยังเขตทิศเหนือ ไหลลงเขตทิศใต้ ไหลไปเขตทางตะวันออก และ ไหลไปเขตทางตะวันตก โดยเขตรับน้ำจะไม่สามารถระบายน้ำกลับไปยังเขตก่อนหน้าที่ระบายน้ำมาให้

จงเขียนโปรแกรมเพื่อคำนวณหาระยะเวลาที่น้อยที่สุด ที่น้ำทิ้งอย่างน้อย 2 สายจะมาบรรจบกัน พร้อมทั้งบอกพิกัดของเขตที่น้ำทิ้งมาบรรจบกัน (รับประกันว่าข้อมูลนำเข้าทุกชุด จะมีเขตที่น้ำสองสายมาบรรจบกันที่เกิดขึ้นเร็วที่สุด เพียงเขตเดียวเสมอ) โดยจากรูปตัวอย่างข้างบนนี้ น้ำทิ้งจะเริ่มตันที่  $(1, 1)$  ในช่วงเวลาที่ 1 และเคลื่อนไปสู่  $(2, 1)$  และ  $(1, 2)$  ในช่วงเวลาที่ 2 จากนั้นจึงไปสู่  $(3, 1)$  และ  $(1, 3)$  ในช่วงเวลาที่ 3 และถึง  $(3, 2)$  กับ  $(2, 3)$  ในช่วงเวลาที่ 4 และสุดท้ายจึงมาบรรจบกันที่พิกัด  $(3, 3)$  ในช่วงเวลาที่ 5 ตามลำดับ

กำหนดให้แต่ละเขตสามารถมีรูปแบบการติดตั้งท่ระบายน้ำได้ทั้งหมด 4 รูปแบบ เมื่อพิจารณาการเชื่อมต่อทางทิศตะวันออกและทิศใต้เท่านั้น ได้แก่ R หมายถึงเขตนั้นมีท่ระบายน้ำเชื่อมกับเขตทิศตะวันออก, D หมายถึงเขตนั้นมีท่ระบายน้ำเชื่อมกับเขตทิศใต้, B หมายถึงเขตนั้นมีท่ระบายน้ำเชื่อมกับทั้งเขตทิศตะวันออกและทิศใต้, และ N หมายถึงเขตนั้นไม่มีท่ระบายน้ำเชื่อมกับเขตทิศตะวันออกและทิศใต้

### ข้อมูลนำเข้า

- บรรทัดแรกเป็นค่าของตัวแปร  $a$  และ  $b$  โดยที่  $2 \leq a, b \leq 100$
- บรรทัดที่สองถึง  $a + 1$  แต่ละบรรทัด มีตัวอักษรทั้งหมด  $b$  ตัว คั่นด้วยช่องว่าง แต่ละตัวระบุถึงสถานะการมีท่อระบายน้ำของเขตแต่ละเขตในพิกัด  $(i, j)$  โดยเริ่มจากพิกัดที่  $(1, 1)$  ไปเรื่อย ๆ ตามลำดับ และ  $1 \leq i \leq a$  และ  $1 \leq j \leq b$

### ข้อมูลส่งออก

- บรรทัดแรกเป็นจำนวนเต็ม 1 ตัว แสดงถึงช่วงเวลาที่น้ำทิ้งมาบรรจบกัน
- บรรทัดที่สองเป็นจำนวนเต็ม 2 ตัว คั่นด้วยช่องว่าง ซึ่งเป็นพิกัด  $(i, j)$  ที่น้ำทิ้งมาบรรจบกัน

#### ตัวอย่างที่ 1

ข้อมูลนำเข้า	ข้อมูลส่งออก
4 4 B R D N D R B D R R R D N N N N	5 3 3

#### ตัวอย่างที่ 2

ข้อมูลนำเข้า	ข้อมูลส่งออก
3 4 B B B D D N R B R R R N	5 2 4

## ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	16 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

## ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล sewer.c และระบุส่วนหัวของโปรแกรมดังนี้

/\*

TASK: sewer

LANG: C

AUTHOR: YourName YourLastName

CENTER: YourCenter

\*/

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล sewer.cpp และระบุส่วนหัวของโปรแกรมดังนี้

/\*

TASK: sewer

LANG: C++

AUTHOR: YourName YourLastName

CENTER: YourCenter

\*/